Content-based UEP: A new Scheme for Packet Loss Recovery in Music Streaming

Ye Wang[1,2], Ali Ahmaniemi[2], David Isherwood[2],Wendong Huang[1]

[1]School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543
wangye@comp.nus.edu.sg

[2]Audio and Visual Systems Laboratory,
Nokia Research Center,
Tampere, Finland
{ali.ahmaniemi, david.isherwood}@nokia.com

ABSTRACT

Bandwidth efficiency and error robustness are two essential and conflicting requirements for streaming media content over error-prone channels, such as wireless channels. This paper describes a new scheme called *content-based* unequal error protection (C-UEP), which aims to improve the user-perceived QoS in the case of packet loss. We use music streaming as an example to show the effectiveness of the new concept. C-UEP requires only a small fraction of the redundancy used in existing forward error correction (FEC) methods. C-UEP classifies every audio segment (e.g. an encoding frame) into different classes to improve encoding efficiency. Salient transients such as drumbeats and note onsets are encoded with more redundancy in a secondary bitstream used to recover lost packets by the receiver. Formal perceptual evaluations show that our scheme improves audio quality significantly over simple muting and packet repetition baselines. This improvement is achieved with a negligible amount of redundancy, which is transmitted to the receiver ahead of playback.

Categories and Subject Descriptors

C. 3. [**Special-Purpose and Application-based Systems**] Signal Processing Systems

H.5.5. [**Sound and Music Computing**]: Signal Analysis, Synthesis and Processing, Systems

General Terms

Algorithms, Performance, Reliability, Experimentation, Human Factors

Keywords

Audio Coding and Streaming, Error Robustness, Content-based Unequal Error Protection (C-UEP), Packet Loss Recovery, User-perceived QoS, Prioritized Resource Allocation

1. INTRODUCTION

The non-real-time transmission of compressed digital audio, such as MP3 (MPEG-1 layer 3), over the Internet has had a profound effect on the traditional process of music distribution. With increasing channel capacity available in the new generation of mobile networks, it is possible to stream compressed media content to wireless mobile terminals via the Internet. However, the characteristics of this scenario pose special problems. A significant challenge is the need for error handling, more specifically packet loss recovery.

Packet loss can arise in many different forms. On the Internet, packets can be dropped due to congestion at switches, they can be misrouted, or they can arrive with such a long delay as to be useless. On wireless networks, packet losses can be caused by channel characteristics, such as fading and interference, or wireless network characteristics, such as handover in a cellular network. Under such conditions, it is crucial to guarantee user-perceived QoS for widespread acceptance of media streaming applications. Our survey of three campuses in Finland, England and Singapore confirmed that the user-perceived QoS is among three dominant factors, along with "interesting content" and "low price" for adopting audio and multimedia streaming services.

The objective of packet loss recovery is to reconstruct a lost packet so that it is perceptually indistinguishable, or sufficiently similar to the original one. This objective should be achieved with minimal system cost. From the resource allocation perspective, the research problem in this paper can be defined as one of prioritized resource optimization based on the following facts. First, once a wireless technology (2G, 2.5G, 3G, Wireless LAN, Bluetooth, etc) is established, its maximum channel capacity is fixed for the lifecycle of the technology. This scarce resource is usually shared by many users. This justifies our special effort to reduce bandwidth consumption in the proposed scheme. Second, the computational power and memory in the terminals (laptop, pocket PC, smart phone, PDA, etc.) are constantly improving according to Moore's law. This increasingly available resource can be used for packet loss recovery. Third, the computational power and memory in the server can be assumed unlimited, especially for offline computations. The proposed scheme is based on the above assumptions.

Unequal error protection (UEP) is an important subclass of forward error correction (FEC). UEP gives more protection to the

more important information bits and enables a better trade-off between the performance and required redundancy than conventional FEC. UEP has been an effective tool for protecting compressed domain audio bitstreams such as MPEG-4 AAC against *random errors*. We categorize the traditional UEP as bitstream-level UEP (B-UEP), which is deployed within individual media frames. Without a smart packetization scheme, it is less effective in packet-switched streaming, because the network typically discards the entire packet if there are bit errors detected. The B-UEP is a low-level UEP, which does not exploit any high-level (semantic-level) structures of media streams.

In this paper, we advocate a semi-semantic-level *content-based* unequal error protection (C-UEP) framework. The proposed framework tackles the unequal error protection problem from a different perspective – content segmentation, classification and prioritization. Only the most significant streams or segments get the highest level of protection. Our scheme is designed with packet-switched networks in mind, and in this paper is proven to be more effective than existing methods in the case of music streaming.

We will show that salient transients such as drumbeats and note onsets in music should be better protected in comparison with their quasi-stationary counterparts as a chord from a synthesizer if we want to achieve good perceptual quality in packet loss recovery with minimum redundancy.

We combine receiver-based error concealment methods and modern parametric/structured audio coding techniques where transients are modeled as elementary objects. This methodology combines the strength of receiver-based error concealment and sender-based FEC, while minimizing their weaknesses.

This paper is organized as follows. After this introduction a brief review of related works is given in Section 2. Then our conceptual framework and methodology are outlined in Section 3, followed by our current implementation of the system in Section 4. Formal perceptual evaluation results are presented in Section 5. Discussions are given in Section 6. Finally, Section 7 concludes the paper.

2. RELATED WORK

There are many published works related to packet loss recovery (see [1][2] for overviews). We classify them into three categories: sender-based, network-based and receiver-based.

Most existing sender-based methods belong to FEC, which can only achieve good performance when a considerable amount of redundant information is sent [1]. One of the initial motivations of this paper was to significantly reduce the redundancy required by traditional approaches in order to match the bandwidth constraints in wireless applications.

Network-based methods are mostly based on re-transmission mechanisms, which have the penalty of long latency and overhead redundancy. In certain applications, such as broadcasting and multicasting, re-transmission is not desirable or even simply not possible. The proposed scheme aims to reduce the need for re-transmission.

Receiver-based error concealment methods serve as the last resort to mitigate the degradation of audio quality when data packets are lost. Error concealment methods generally exploit correlations between adjacent packets. The use of packet repetition is recommended as offering a good compromise between achieved quality and complexity [2]. However, these types of methods rely on assumptions that packet loss is infrequent, that the packet size is small and that the signal is fairly stationary. The last assumption is the basis for most existing methods and is not always valid, especially in the case of transients in music. Using this approach, it is very difficult, if not impossible to guarantee the user-perceived QoS in the case of packet loss. One of the key contributions of this paper is how to improve the user-perceived QoS in the case that the lost packet is close to a transient.

Most published works in audio packet loss recovery have focused on speech [1][2][3]. Relatively few published works have dealt with high quality music streaming. It is worth mentioning that speech is mostly used for communications, while music is used for entertainment. Because of their different purposes, their quality requirements in streaming applications are quite different. In general, quality requirements for music are more demanding than for speech, partially due to the high expectations after three decades of exposure to CD quality audio. In addition, the primary objective in recovering speech signals is intelligibility, not that of audio quality. The different objectives and signal characteristics affect the choice of the optimal algorithm for packet loss recovery.

It should be noted that different users have different expectations and requirements on QoS in music streaming. Our experience shows that small impairments introduced by a perceptual codec such as MP3 are acceptable for the majority of the general public. However, disruptive impairment, which is introduced by naïve error recovery techniques such as muting and simple packet repetition, can be irritating and unacceptable.

To achieve sufficient error robustness in streaming high quality audio, we have tried to adopt existing schemes such as the media-specific forward error correction (FEC) scheme presented in [2], which is illustrated in Figure 1.

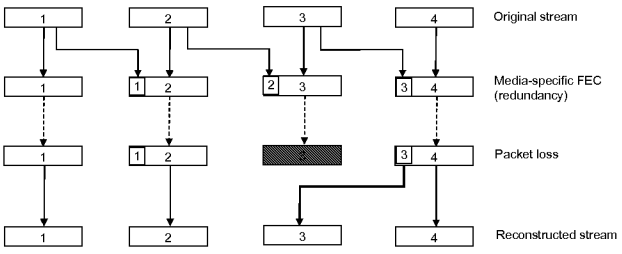

Figure 1. Concept of error recovery technique using a secondary bitstream in addition to the primary bitstream.

In the course of our research, we have found some problems in the above concept for music streaming applications.

- Existing methods require too much redundancy. If we use MPEG Advanced Audio Coding (AAC) as the primary encoding with 64 kbps and Adaptive Multi-Rate Wideband (AMR-WB) [4] as the secondary encoding with 16 kbps, a 25% overhead is incurred. This overhead is incurred for every packet, protecting against occasional packet loss. Based on our investigations, this is not the best option in a

wireless environment. Our solution is to employ a novel structured music encoding scheme to significantly reduce the needed redundancy, which is delivered reliably and stored in the receiver to recover possible lost packets. This approach increases its robustness against burst packet loss.

- Different codecs have different frame sizes. This sets some extra constraints on what codec can be used in a particular scheme [5]. In addition, it is not desirable to run two complex decoders for both the primary and secondary bitstreams in a small mobile terminal from the perspective of resource consumption. Our solution is to encode the secondary bitstream with a finer time index than the frame size of the primary bitstream. Conceptually this is equivalent to providing time stamps, which guide the receiver to reconstruct the lost packet. Our secondary bitstream is very simple to decode, in contrast to other solutions.
- If one uses a Modified Discrete Cosine Transform (MDCT) domain codec (such as MP3 or AAC) as the primary encoding, and a time domain codec (such as AMR-WB) as the secondary encoding, some special handling has to be performed to avoid the un-cancelled time domain alias and block effect [6]. Otherwise, the secondary bitstream may be rendered uselessly. That is, the result of using the secondary bitstream can be worse than a simple receiver-based error concealment method in case of packet loss. We exploit the unique characteristics of MDCT [6] in our solution.
- In FEC, the correlation between neighboring packets has not been utilized. In addition, the concept in Figure 1 is less effective in the case of burst packet loss. Our solution is to fully exploit the inter-packet correlation to recover quasi-stationary components in music.

The proposed scheme, which addresses the above-mentioned problems, aims to significantly reduce the redundancy incurred in FEC and the overhead in network re-transmission, yet to achieve much better perceptual audio quality in comparison with receiver-based error concealment methods. This is achieved by shifting the most demanding computation to the sender side and by exploiting the increasingly available computational power and memory capacity in the mobile terminals.

Our initial idea of utilizing musical beat structure to recover packet loss was presented in [7] and [8]. A receiver-based error concealment approach was presented using MP3 audio bitstreams and its performance was limited in practical applications. Subsequent partial progress has been reported in [9] and [10] using AAC audio bitstreams. This paper presents our overall conceptual framework and reports our latest developments and findings.

3. CONCEPTUAL FRAMEWORK

The presented framework is a combination of FEC and receiver-based error concealment.

The receiver-based error concealment approach (interpolation, extrapolation, etc.) does not require any redundant packet-sending from the sender. In our experiments we found that the user-perceived QoS of the reconstructed audio signal generally depends on the characteristics of the signal – error concealment usually works well if the signal is quasi-stationary but suffers badly if the lost packet is close to a transient such as a drumbeat or note onset. In addition to the double-drumbeat effect [9], one can experience a distortion, which we define as the *melody-disruption-effect*, if a simple packet repetition is used to recover a lost packet around a note change. The *melody-disruption-effect* is particularly noticable in music that does not have percussive rhythm. Classical music exhibits this property. This effect is illustrated in Figure 2.

Based on the above observation, it is clear that the result of error concealment can be rather poor if the structure of the music is not effectively utilized. This gives us a clear clue as to how to solve the problem – we simply need to indicate the locations of all transients and their key attributes. That is, the performance of error recovery can be significantly improved with structural knowledge of music signals.

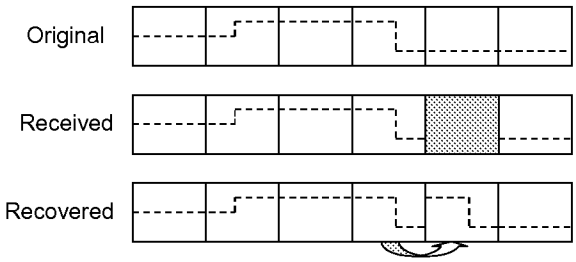

Figure 2. Melody-disruption effect with a simple packet repetition approach. The blank rectangle represents a correctly received packet, and the shaded rectangle represents the lost packet. The dashed line illustrates the pitch of the music signal.

In conventional media-specific FEC, the secondary bitstream (redundancy) is just a degraded version of the primary bitstream. It does not utilize the correlation between adjacent packets [1], or music structure. This explains why the conventional method is not efficient.

In order to reduce the secondary data significantly, we focus on a compact representation of the salient transients in music, which guides the receiver to reconstruct lost packets around transients with better results. For the quasi-stationary part, we simply rely on the receiver-based error concealment based on the correlation between adjacent packets. That is, the *only* redundancy is transmitted from the sender in advance, where the receiver cannot recover the lost packet from its neighboring packets. This approach eliminated the problems of both the conventional receiver-based error concealment (unpredictable quality) and the conventional FEC (excessive redundancy).

What are the salient transients in music? Although different people may have different opinions, we broadly classify the salient transients of music into two categories as illustrated in Figure 3. For simplicity we limit the salient transients to include only drumbeats (short bursts of sound) and note onsets that rise to full intensity from a low level followed by little or no decay.

Figure 3 shows a possible classification tree for coding a music segment. More detailed characteristics are given as the tree grows. At the second level there are four classes: symbol (a) represents silence, (b) stationary sound, (c) drumbeat, and (d) note onset without drumbeat. At the third level, some classes can be split

further. Symbol (c1) represents a drumbeat without other sustaining sound, (c2) drumbeat with other sustaining sound such as singing, (c3) drumbeat associated with note onset.

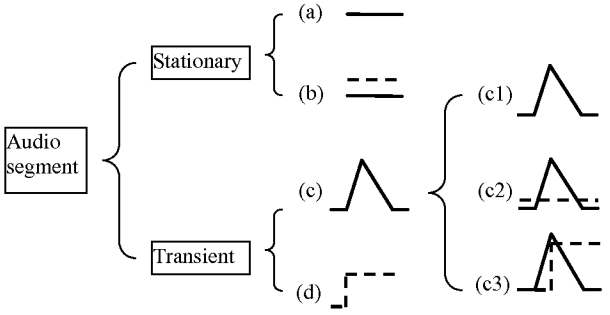

Figure 3. Classification of music segment – a symbolic representation. The solid lines and dashed lines represent loudness and pitch respectively. Letters indicate different classifications of the segment (see details in the main text).

All the leafs of the classification tree are encoded with the key attributes, such as a pre-classification index, the onset position. The encoded secondary bitstream is sent to the receiver for packet loss recovery.

Although the scheme illustrated in Figure 1 can be used for delivering the secondary bitstream, it is not the best operation mode for two reasons. First, the scheme prevents us from using the most compact encoding method to represent the secondary bitstream. Second, it is not effective against burst packet loss. Therefore, we send the secondary bitstream of an entire song as a chunk to the receiver using a reliable transmission mechanism. This is done in parallel with buffering before the playback at the receiver begins. The transmission of the small amount of secondary data increases the buffering time slightly.

This secondary bitstream is then decoded and stored in the receiver to repair possible packet loss. It will be up to the constraints of the terminal's computational and memory capacity how the secondary bitstream is used to perform the error recovery.

4. SYSTEM IMPLEMENTATION

The current implementation of our system is divided into two parts: structured music encoding offline at the sender side and packet loss recovery operation at the receiver side. The proposed concept can be implemented with any specific audio codec. Our current implementation is based on MPEG AAC frame structure. That is, we use an AAC bitstream as the primary bitstream and our structured encoding scheme as the secondary bitstream.

4.1 Structured Music Encoding at the Sender Side

A high-level block diagram of the proposed transient encoding scheme is illustrated in Figure 4.

The first step is an onset detector, which picks up all salient transients with an AAC sub-frame accuracy (~ 3 ms in time resolution) [10]. Then, these onsets are characterized and classified as either drumbeats or note onsets without drumbeats (see Figures 3 and 4). These classes are encoded differently. In our current implementation, our classification tree stops at the second level.

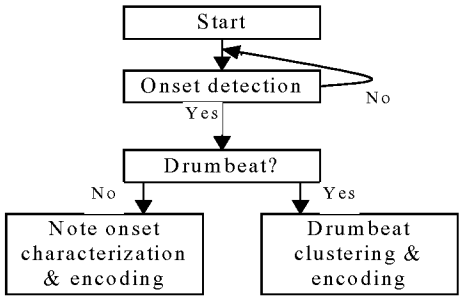

Figure 4. Onset detection, classification and encoding.

Individual blocks of the structured music-encoding algorithm are explained in the following subsections.

4.1.1 Onset Detection

The onset detector is similar to our earlier system in [10], which detects onsets based on intensity change in subbands. Some modifications are made according to [11] in order to pick up softer onsets. The key features of our onset detector are summarized in this sub-section.

We limit the maximum number of onsets within each AAC frame to one. There are 8 short windows within each AAC frame, as illustrated in Figure 5. The time duration of an AAC frame is approximately 46 ms for the sampling frequency of 44.1 kHz. In consideration of the 50% window overlap, the time resolution of the onset detector is roughly 3 ms, which is sufficient for monophonic audio signals [16].

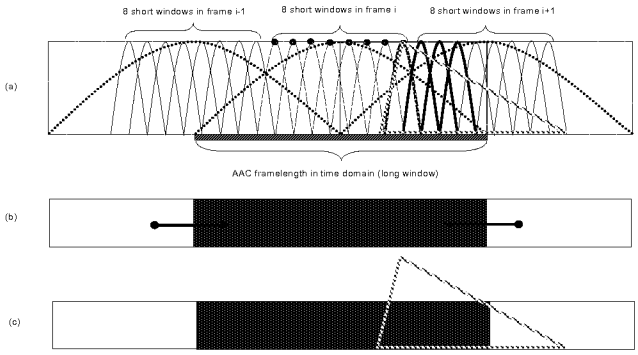

Figure 5. Improved time resolution for onset detection (a) and lost frame reconstruction (b-c). The shaded rectangle represents the lost AAC frame. The two arrows represent interpolation. The triangles represent drumbeats.

Figure 5(a) illustrates AAC frame structure and short windows. The 8 dots indicate the central positions of 8 short windows, indicating finer time grids. (b) illustrates the reconstruction of the missing stationary objects based on interpolation. (c) illustrates the reconstruction of a frame having drumbeat. The lost frame is first reconstructed by a band-limited interpolation, and then mixed with a stored drumbeat.

For onset detection in subbands, there are two essential components: feature extraction and threshold-setting. We use the same features as in [10], but modified our adaptive threshold to:

$$F_{thr} = m + k \cdot std + C, \quad (1)$$

where m and std are the mean and standard deviation respectively that are calculated over a long rectangular window of 301 short windows (~ 900 ms) excluding the middle 5 short windows (see Figure 6). The reason for excluding the 5 middle short windows is that we want to increase the probability to pick up an impulsive candidate, thus reducing the probability of missing an onset. k is a constant that determines the percentage of selected candidates over the total number of candidates. C is a constant that prevents the threshold getting too low, and indicates the minimum detectable changes in each subband. It is calculated based on a large set of training data statistics.

Figure 6 shows different windows and their relative positions for onset detection and subsequent classifications. For onset detection, we use short overlapping windows (solid lines in Figure 6) to extract features with good time resolution (~3ms). For subsequent transient classifications, we use long sine windows (dashed lines in Figure 6) to extract features with an increased frequency resolution. The arrow indicates a salient onset and the current time index. The long rectangular window is used in threshold-setting for onset detection.

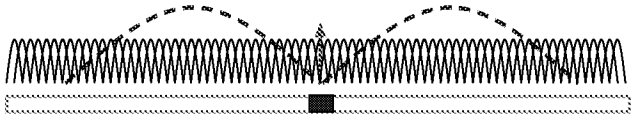

Figure 6. Onset detection and classifications using different window shapes and sizes.

Figure 7 shows an example of how an onset is detected using our subband approach. The waveform in the time domain is illustrated in (a). Feature vector (FV) in subbands 4-1 with thresholds (generally horizontal lines) are illustrated in (b-e). The detected onsets are indicated in (f).

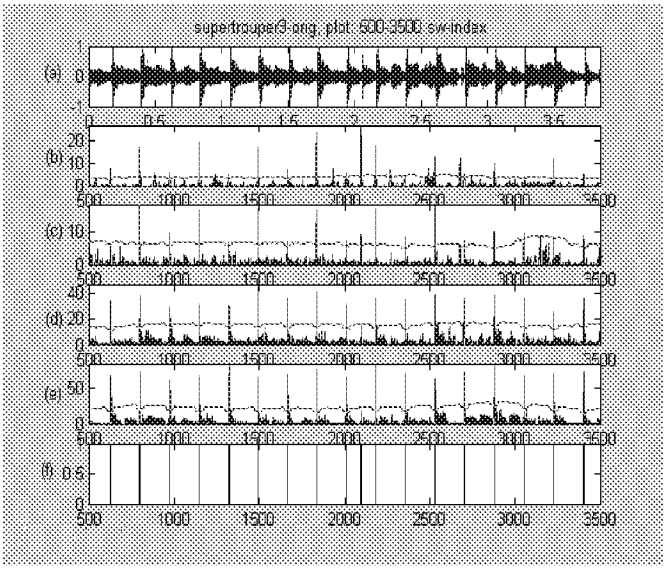

Figure 7. Onset detection in subbands. (a) waveform versus time index in second, (b-e) subband features and thresholds versus short window indices, (f) detected salient onsets versus short window index.

Drumbeat detection in pop-music is a relatively straightforward task, in contrast to note onset detection. Due to characteristics of different instruments, note onset analysis can be a challenging task even for an experienced musician. Some instruments, such as the violin, have relatively long attack time and note onsets are not necessarily seen as peaks in energy domain. Also, the concept of a note onset is not as well-defined as a drumbeat; some instruments can slide from one note to another, in which case the exact position of a note change is impossible to define. However, we focus on salient note onsets, which are relevant for our application.

4.1.2 Structured Music Encoding

For the sake of simplicity, we employ a decision tree to classify and subsequently encode every AAC frame. With salient onset detection, every AAC frame is first classified into two classes: stationary and transient. At the second level, stationary frames are further classified into silence and sustaining sound, while transient frames are further classified into note onset without drumbeat and drumbeat. In the conceptual level, our encoding scheme can be considered as a special implementation of that in [15].

We use a linear classifier to classify a stationary AAC frame further into silence or sustaining sound based on a simple threshold of the sound intensity of the frame. Transient frames are more complicated to deal with. For simplicity, we classify them into two classes: note onsets and drumbeats. Therefore, we need only 2 bits to pre-classify every AAC frame as shown in Figure 8.

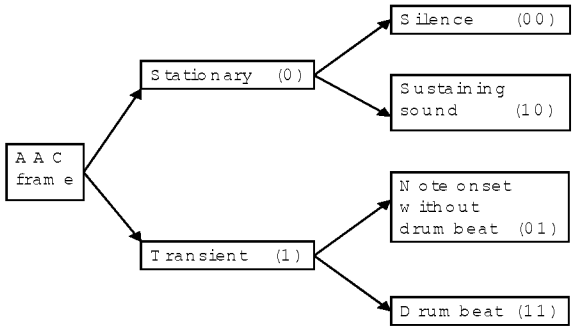

Figure 8. Pre-classification of every audio segment (an AAC frame) and encoding with 2 bits.

For a transient frame, a few more bits are needed to encode its key attributes, such as onset time index. In our current implementation, 3 bits are used for encoding the time index of the onset within each AAC frame (see Figure 5(a)). However, it is a challenging task to determine the optimal amount of bits for encoding other attributes of drumbeats and note onsets.

The drumbeats are encoded with a parametric vector quantization (PVQ) scheme as outlined in [10]. PVQ is used to cluster drumbeats into a few classes due to the highly repetitive nature of drumbeats in music. Drumbeats are commonly used in pop music to maintain musical beat and are generally difficult to reconstruct

from neighboring frames using interpolation or extrapolation. This explains why it is necessary to transmit a small amount of audio samples in the form of a codebook from the sender to the receiver [10]. The codebook in this paper means a set of short audio segments in the PCM domain, which are the representatives of all drumbeats in a song. It is fairly straightforward to see that a drumbeat without other sustaining sound is usually the best candidate for the percussive codebook (see Figure 3 (c1)).

Intuitively, the VQ codebook size should be the number of different drumbeats and their combinations used in a piece of music. It is rather difficult to determine the right size of codebook since the number of different drumbeats in a piece of music is unknown. However, the purpose of our PVQ is not to distinguish different drumbeats, but to cluster them into a number of artificial classes based on their perceptual similarity [10]. Using expectation maximization (EM)-based algorithm, the total average-distortion of VQ will decrease monotonically when the size of codebook increases. That is, the larger the codebook size, the more accurate representation of the codebook for the FV space.

However, using an unnecessarily large codebook requires time-consuming codebook training and more bits for encoding the codebook and codeword index. Our experiments have shown that 8 clusters are sufficient for most of our test music signals. Even 4 clusters can be satisfactory for a large number of music samples. The drumbeats within each class are perceptually similar [10].

Note onsets without drumbeats are more common in classical music, where the melody is a key attribute. Since the signal before and after a note onset is quasi-stationary, it usually requires a smaller amount of data for the structured encoding in comparison to a drumbeat. In our current implementation, we use 3 bits for encoding the position of the onset and 3 bits for encoding the slope of note onset.

These are the general principles of our transient encoding scheme. Based on our experiments, a majority of AAC frames in our test music signals are quasi-stationary. Therefore only 2 bits redundancy is needed for most parts of a music signal. For the transient segments, different coding methods apply as described earlier. An example is given to show a possible secondary bitstream in Figure 9.

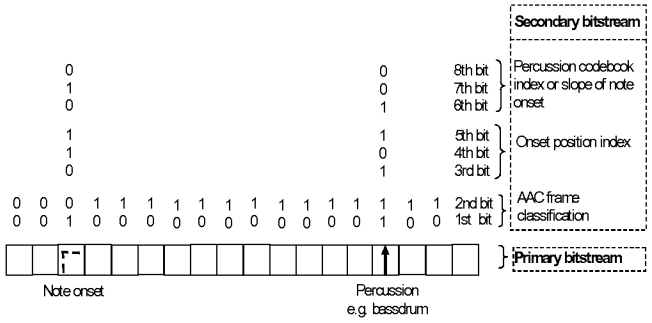

Figure 9. Data structure of primary and secondary bitstreams. The solid rectangles represent AAC frames (primary data). The arrow represents a drumbeat. The step-function represents a note onset.

To distinguish drumbeats from note onsets, we use three features: 1) the temporal energy contours, especially the onset and decay slopes in the time domain; 2) the spectral flatness measure (SFM) [24] in the frequency domain, for note onsets usually have clear harmonic structure, while drumbeats are quite chaotic with rather flat spectrum; 3) the bandwidth defined in [14], since the bandwidth of a note onset is usually smaller than that of a drumbeat. We use the above features to form a 4-dimensional feature vector (2 dimensions for temporal energy contour, 1 for spectral flatness, and 1 for bandwidth), and employ the same LBG-VQ algorithm as in [10] to classify the feature space into two classes. It is clear that the above feature space is different in comparison to that for onset detection.

4.1.3 Transmitting the Secondary Bitstream

The general principle of transmitting some redundancy (media-specific FEC) in a separate packet is not novel and was discussed in [3]. In the conventional approaches, the primary and secondary bitstreams are transmitted with the same mechanism and priority (see Figure 1).

In our scheme, the payload of each packet is an AAC frame (~46 ms), which serves as a gross time index. The finer time index is provided by the 8 short windows within each AAC frame (see Figure 5). Using the same time index, it is easy to synchronize the primary and secondary bitstreams during playback and packet loss recovery. With this timing mechanism, we can transmit the primary and secondary bitstreams separately with different priority and robustness. In our scheme, we transmit the entire secondary bitstream including the percussive codebook ahead of the playback. In other words, we assume error-free transmission of the secondary bitstream.

This transmission mode enables us to use a more compact encoding scheme such as a run-length encoding algorithm, since there are usually long quasi-stationary segments between two adjacent transients. The run-length encoding algorithm is briefly explained with help of Figure 9. We use a symbol to represent a codeword of the secondary data. Each symbol indicates the same gross time index as an AAC frame. Assume that *A* represents 00 (silence) and *B* represents 10 (quasi-stationary). If there are long sequences of *A* or *B*, the run-length encoding can greatly improve the coding efficiency.

The above transmission mode is possible due to the extremely small amount of secondary data. It does not result in long delay in buffering even on band-limited wireless channels. Assume that the duration of drum clips in the drumbeat codebook is 2048 PCM samples and the codebook size is 4, it results in 16 Kbytes of data. If a music signal is 5 minutes in duration and has 4 transients/second on average as in one of our test samples, the secondary bitstream for indexing is approximately 2 Kbytes. The total secondary data in this case is 18 Kbytes, which increases the buffering time by about 200 ms. In contrast, a conventional FEC using a secondary encoding with 16 kbps results in 600 Kbytes of data. Our encoding scheme consumes 3% of the bits needed for the conventional FEC, which is a significant saving.

4.2 Recovering Lost Audio Packets at the Receiver Side

From the secondary bitstream (semi-semantic metadata), the receiver knows the characteristics of every AAC frame. Based on

the different characteristics of the lost AAC frame, the receiver can choose various strategies for lost packet recovery.

In the receiver, the secondary bitstream is decoded and stored in a buffer before the actual streaming begins. If a packet loss is detected, (e.g., via sequence number of a packet), the receiver will check the stored secondary data for the signal characteristics. If the lost packet and its immediate neighboring packets are quasi-stationary, it uses conventional error concealment methods such as repetition to reconstruct the lost packet. If the computational complexity and memory consumption is not an issue, more sophisticated methods such as the ones in [12] and [13] can be employed.

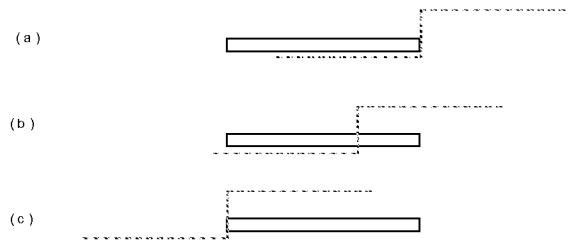

Figure 10. Relative location of the lost packet (blank rectangles) and the pitch (dashed lines).

If the lost packet is close to a note onset as illustrated in Figure 10, the lost packet recovery should be performed according to the relative locations of the two. In the case of (a), the lost packet should be extrapolated from the previous packet. In the case of (b), the lost packet should be extrapolated from both sides until the note onset. In the case of (c), the lost packet should be extrapolated from the following packet.

To minimize a possible blocking effect on the boundary, it is generally necessary to have a suitable cross-fade.

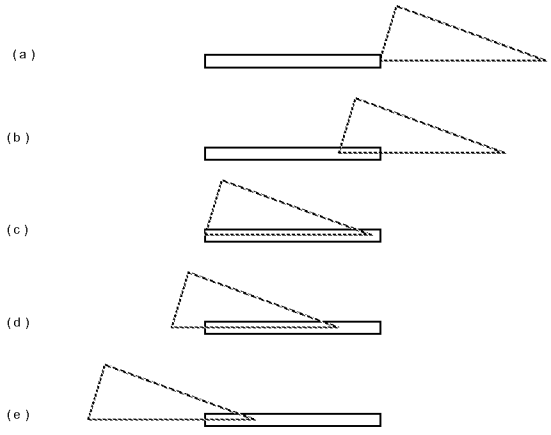

Figure 11. Different packet loss recovery operations depending on the relative locations of the lost packets (rectangles) and the percussions (triangles)

If the lost packet is close to a drumbeat as illustrated in Figure 11, the receiver should reconstruct the lost packet also according to their relative locations, which critically affect the reconstruction performance. In the case of (a), the lost packet should be reconstructed only using the previous packet to avoid the *double-drumbeat-effect*. In the cases of (b) and (c), where the onset of the percussion is within the lost packet, and it will be wise to use both the immediate neighboring packets and the drumbeat codebook to reconstruct the lost packet. Interpolation of the neighboring packets is used to reconstruct the stationary component, which is then mixed with the correct drumbeat from the stored codebook as illustrated in Figure 5 (b and c). In the case of (d), the lost packet is directly after the onset. It is advantageous to use simple interpolation between the previous and the following packets in the frequency domain, but without using the buffered drumbeat to avoid the *double-drumbeat-effect*. In the case of (e), the lost packet should be reconstructed using its subsequent packet.

A simplified formulation of the mixing in time domain is as follows (see Figure 5);

$$x_i = \beta(\alpha x_{i-1} + (1-\alpha)x_{i+1}) + (1-\beta)p_j \qquad (2)$$

where i is the sequence number of packets, x is time domain samples of a packet, p_j is a drumbeat selected from the codebook, α is a cross-fade function to avoid possible discontinuity of the reconstructed stationary component [12], β is cross-fade function for mixing the percussion. β models the contour of the percussion. For simplicity, β can be a simple triangle function to model the contour of a percussion as shown in Figure 11.

5. PERCEPTUAL EVALUATIONS

During the development of the C-UEP algorithm, continuous benchmarking against different recovery methods was done. Of these benchmarking sessions, some were informal tests performed by the authors while others were formal listening tests performed by a battery of experienced listeners to try to find not only the relative ranking between methods but also the perceptual distances qualitatively separating them. These formal tests were designed and administered to validate the author's experience that C-UEP offered superior audio quality in the case of packet loss near percussive segments when compared with conventional methods such as muting and packet repetition.

To highlight our algorithm's performance around transient packets, tests were designed to have one packet loss around every transient, which translates to packet loss rate in the range of 2-5%. The codebook size of the PVQ is 4.

This section describes one such formal listening test. At the point during the development process that this test was performed the onset detector was primarily designed for detecting drumbeats, and was not capable of detecting pure note onsets produced by e.g. violin. Nevertheless, the results still provide a relevant measure of how well C-UEP performs.

5.1 Test Stimuli

A comparison between four audio streaming scenarios was decided upon. The scenarios were; 1) no lost packets [ORIGINAL], 2) lost packets are replaced using content-based unequal error protection [C-UEP], 3) lost packets are replaced with the previous error free packet [REPEATED] and 4) lost packets are muted [MUTED].

The new C-UEP scenario is described in detail in the previous sections. With REPEATED scenario, the lost frame is simply replaced with the previous frame. With MUTED scenario, all 2048 PCM samples within each missing frame are set to zero.

Programme	Time signature	Tempo (qpm)	Description
Slow Rock	4/4	81	Distorted guitar, bass, piano and drums wt melodic instruments sustaining held chord. Dynamics dominated by drums. No vocals.
Dance	4/4	123	Electronic dance music with prominent female vocal through entire programme. Consistent "disco" type bass drum on every beat.
Prog. Rock	9/8	112*	Progressive rock with drums playing polyrhythms with unconventional use of accented individual drums. No vocals.
Country	4/4	120	Prominent strummed acoustic guitar, female vocals and laptop slide guitar. Guiro is also played throughout with drums less prominent

* Specified tempo is equivalent quarter note tempo

Table 1. Description of programmes used in listening test.

Four musical programmes, which have a broad range of musical and acoustic functions, were chosen to evaluate our new recovery method. All four programmes were between 20 and 30 seconds in duration. Table 1 gives some details of the properties of these. In general, these were chosen due to having differing tempos, dynamics and accents. Programmes were also chosen to have greater or lesser melodic/harmonic sustain, e.g. having a sustained vocal performance that is varying melodically. These choices were made to stress the C-UEP method in a range of ways.

5.2 Test Design

A forced-choice, binary paired comparison test methodology was chosen for the test, for which the listener is presented with two stimuli that are the same program material but with different recovery methods applied. The listener is then forced to simply choose, "Which of A and B do you think has better reproduction quality". The paradigm states that if the stimuli are qualitatively equivalent then there will be equal occurrence of preference for both over all presentations. The statistical significance associated with the proportions of preference for both stimuli can state whether one stimulus is preferred over the other [18]. This was felt to be the most suitable test methodology because it is simple for the listener to comprehend and use, is likely to result in less noisy data than scaled paired comparison methodologies when listeners that are unfamiliar with critically differentiating certain types of stimuli are used, and produces data having a comprehensive family of statistical analysis methods that can be applied to it.

For the comparison between the 4 streaming scenarios, presentation of all permutation pairs (A-B and B-A) results in 12 different stimuli pairs for each program. Each of these pairs was presented 4 times to each listener, resulting in 192 test pairs that each listener had to grade. Additionally, 24 pairs, representing all combinations of scenario pairs for all programs, were presented prior to testing as training in the use of the test method and user interface, and to familiarize the listeners with the stimuli.

Thirteen listeners that were experienced in listening to streamed audio over the Internet, and thus were familiar with packet loss artifacts, were chosen to perform the test. All had experience in performing listening tests on perceptual encoders and had proven to have good intra-rater reliability. However, given their naivety in performing listening tests involving packet loss, they could not be classified as expert listeners *a priori* based on previous inter-rater agreement. All were males in their 20s with no hearing loss.

The test was administered using the Guineapig listening test system [19] in a controlled, silent listening environment, specified in [21]. The audio signal chain for presentation is shown in Figure 12, and the user interface used for presentation of pairs is shown in Figure 13. The stimuli were 16bit, 44.1kHz PCM recordings of the original material. The monophonic stimuli were presented diotically over headphones. All stimuli were loudness aligned using Moore's steady-state loudness model [20] to be 20 sones when averaged across the entire sample. This alignment was performed to negate any biasing effect associated with the loudness of one error recovery method over another.

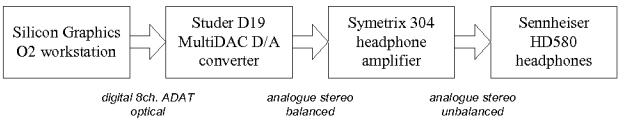

Figure 12. Audio signal chain for presentation of test stimuli.

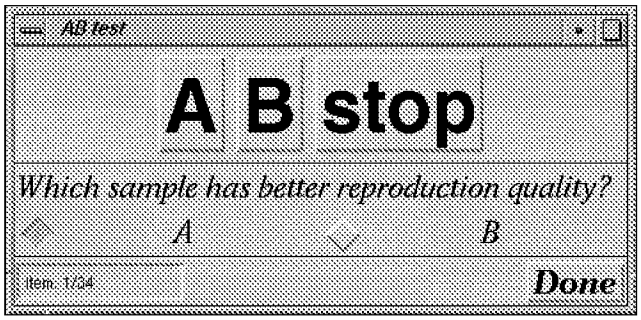

Figure 13. Test administration user interface.

5.3 Test Results

Table 2 contains the proportions of grades given between each streaming scenario; for each program separately, and averaged over all programs. The table is oriented to show preference for each scenario (columns) compared to each of the other scenarios (rows).

A Wilcoxon non-parametric significance test was performed on the data. There was found to be no significant difference between programs ($p>0.05$). The differences between each of the streaming scenarios were all found to be significant ($p<0.01$) for all programs.

The ordinal responses from a test such as this can be converted into interval perceptual distances using a discriminant model based on Thurstone's law of comparative judgment [22]. However, the basic model assumes that grades for a given stimulus are normally distributed, and requires that at least some stimuli overlap measurably, i.e. if the discriminal distribution of any streaming scenarios do not overlap with any of the others, its interval location cannot be determined [23]. In the above data, the ordinal proportions are so polarized that this model cannot be used. This does not infer any shortcomings in the data, rather that the discriminant model cannot be applied when the strength of preference between pairs is so absolute. Thus, only the averaged

proportions for each streaming scenario are calculated, shown in Figure 14.

		Scenarios			
	Scenarios	1	2	3	4
Slow rock	1. Original	-	0.058	0.000	0.000
	2. C-UEP	0.942	-	0.000	0.000
	3. Repeated	1.000	1.000	-	0.038
	4. Muted	1.000	1.000	0.962	-
Dance	1. Original	-	0.173	0.000	0.019
	2. C-UEP	0.827	-	0.000	0.000
	3. Repeated	1.000	1.000	-	0.107
	4. Muted	0.981	1.000	0.893	-
Prog. rock	1. Original	-	0.115	0.000	0.000
	2. C-UEP	0.885	-	0.000	0.000
	3. Repeated	1.000	1.000	-	0.019
	4. Muted	1.000	1.000	0.981	-
Country	1. Original	-	0.173	0.000	0.000
	2. C-UEP	0.827	-	0.000	0.000
	3. Repeated	1.000	1.000	-	0.038
	4. Muted	1.000	1.000	0.961	-
ALL	1. Original	-	0.130	0.000	0.005
	2. C-UEP	0.870	-	0.000	0.000
	3. Repeated	1.000	1.000	-	0.043
	4. Muted	0.995	1.000	0.957	-

Table 2. Proportions of listeners preferring each streaming scenario (columns) compared to each of the other scenarios (rows) for each program separately and averaged over all programs.

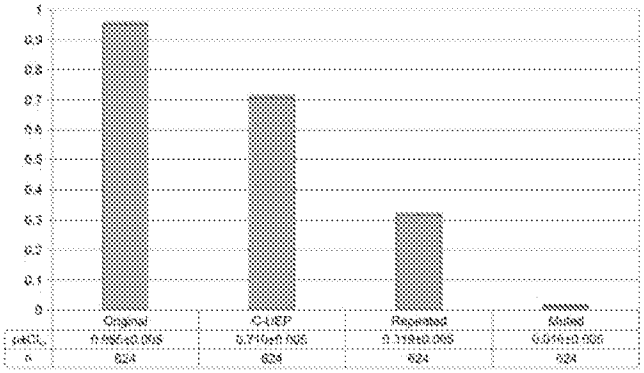

Figure 14. Averaged proportions for streaming scenarios. Significant differences between ranks are found at $p<0.01$

6. DISCUSSIONS

The listening test has shown that C-UEP error recovery is preferable over both repetition and muting for this sample set. Two aspects are discussed in the following sub-sections. The first aspect concerns the strengths, weaknesses, and possible improvements of the scheme in music streaming. The second aspect concerns a generalized C-UEP concept, which we believe is useful in many real-life multimedia streaming applications.

6.1 Strengths, Weaknesses and Possible Improvements

To our knowledge, this is the first attempt to establish a content-based UEP framework for error-resilient music streaming applications with promising results. Our structured music coding scheme is more efficient than the existing coding methods that we are aware of. This framework is particularly suitable for streaming high quality music from a server to many wireless mobile clients, where burst packet loss may happen. However, the particular approach presented in this paper is not suitable for two-way real-time communications due to the requirement of offline encoding in the server.

In principle, a percept of an onset is caused by a noticeable change in intensity, pitch and timbre of the sound [17]. We have only considered onsets caused by intensity and pitch in our current implementation. This is based on our assumption that drumbeats and note onsets are the most common and important musical transients, which are difficult to recover from their neighboring packets. Other musical attributes, e.g. vibrato or gradual fade-in, can be recovered relatively well using receiver-based error concealment.

The structured music data (metadata) is transmitted to the receiver and it is up to the receiver to decide how to use this data. In addition to packet loss recovery, it can also be used for many other purposes such as synchronizing events with music, audio classification and summarization, etc.

In the current implementation, the metadata is transmitted first followed by the RTP stream. This is not an essential requirement. Alternatively, a TCP connection could be used in parallel with the primary RTP stream, or the metadata could be protected with an error correction scheme such as retransmission. These alternatives would only apply to the metadata.

The proposed scheme is suitable for many different types of mobile terminals with different computational and memory capacities. It can also be designed to be independent of any particular audio codec.

The proposed C-UEP scheme can be extended to encode transients other than drumbeats and note onsets. In general, a heterogeneous media stream can be segregated into piecewise homogenous segments. The boundaries between individual homogenous media segments are important. Further research is needed in this direction.

6.2 Generalized C-UEP

To improve user-perceived QoS, UEP is a simple and effective concept which can be deployed at different levels, from lower-level B-UEP to high-level C-UEP, to protect semantically significant parts of multimedia content. C-UEP can also facilitate scalability in streaming applications. That is, in case of bandwidth constraints, packets can be dropped according to their semantic importance.

We believe that different media streams (e.g., audio and video), even different segments in individual media streams, are of different level of importance in streaming services. For example, during the live reporting of breaking news events, if we lose the video track, the service can still be continued with some constraints. However, if we lose the audio track, the service breaks down immediately since it is extremely difficult for us to read lips from the video without an audio track. From an information theory viewpoint, the reporter's face in the video does not convey much information, but his/her report (audio track) conveys the most relevant information to the audiences. Instead of having bad quality in both media streams simultaneously, it can be a better option to freeze the video temporarily and to guarantee the audio stream.

Therefore, it is important to analyze the significance of individual media streams and segments. Based on this information, it is

possible to optimize the resource allocation in different situations and to achieve the best user-perceived QoS. Some efforts in this direction can be found in [25][26].

7. CONCLUSION

A novel content-based unequal error protection (C-UEP) scheme has been proposed for music streaming, which yields a good balance between user perceived QoS and relevant resources. The key technology is to fully exploit the structural characteristics of music signals and encode only the most relevant attributes of transients in a secondary bitstream. In comparison to traditional bitstream-level unequal error protection (B-UEP) schemes, we believe that C-UEP concept is a big step forward in improving user-perceived QoS in many multimedia streaming applications where the bandwidth is constrained.

8. REFERENCES

[1] Wah, B.W., Su, X. and Lin, D., "A Survey of Error Concealment Schemes for Real-time Audio and Video Transmissions over the Internet," IEEE International Symposium on Multimedia Software Engineering, Taipei, Taiwan, pp.17-24, Dec. 2000

[2] Perkins, C., Hodson, O., Hardman, V., "A Survey of Packet Loss Recovery Techniques for Streaming Audio," IEEE Network, pp.40-48, Sept/Oct, 1998

[3] Hardman, V. et al., "Reliable Audio for use over the Internet," Proc. International Networking Conference (INET'95), Hawaii, USA, pp.171-178, June 1995

[4] 3rd Generation Partnership Project, "TS 26.190 V5.0.0, AMR Wideband Speech Codec," 2001

[5] Vilermo, M., et al., "Perceptual Optimization of the Frequency Selective Switch in Scalable Audio Coding" AES 114th convention, Amsterdam, March 2003

[6] Wang, Y., Vilermo, M., "Modified Discrete Cosine Transform – Its Implications for Audio Coding and Error Concealment," Journal of Audio Engineering Society, Vol.51, No. 1/2, pp.52-62, January/February 2003

[7] Wang, Y., "A Beat-Pattern based Error Concealment Scheme for Music Delivery with Burst Packet Loss", IEEE International Conference on Multimedia and Expo (ICME2001), pp.73-76, Tokyo, Japan, August 2001

[8] Wang, Y., Vilermo, M., "A Compressed Domain Beat Detector using MP3 Audio Bitstreams", The 9th ACM International Multimedia Conference (MM2001), Ottawa, Canada, September 30 – October 5, 2001

[9] Wang, Y., Streich, S., "A Drumbeat-Pattern based Error Concealment Method for Music Streaming Applications," International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP2002), Orlando, Florida, USA, pp.2817-2820, May 13-17, 2002

[10] Wang, Y., Tang, J., Ahmaniemi, A., Vaalgamaa, M., "Parametric Vector Quantization for Coding Percussive Sounds in Music," ICASSP2003, pp. 652-655, Hong Kong, China, 2003

[11] Duxburg, C., Sandler, M., and Davies, M., "A Hybrid Approach To Musical Note Onset Detection", Proc. of the 5th Int. Conference on Digital Audio Effects (DAFx-02), Hamburg, Germany, pp.33-38, September 26-28, 2002

[12] Niedzwiecki, M., Cisowski, K., "Smart Copying – A New Approach to Reconstruction of Audio Signals," IEEE Transactions on Signal Processing, pp.58-63, Vol. 49, No. 10, October 2001

[13] Kauppinen, I., Kauppinen, J., Saarinen, P., "A Method for Long extrapolation of Audio Signals," Journal of the Audio Engineering Society, Vol. 49, No. 12, pp.1167-1180, December 2001

[14] Wold, E., Blum, T., Keislar, D., Wheaton, J., "Content-Based Classification, Search, and Retrieval of Audio," IEEE Multimedia Vol.3, No. 3, pp.27-36, Fall 1996

[15] Scheirer, E. D., "Structured Audio, Kolmogorov Complexity, and Generalized Audio Coding," IEEE Transactions on Speech and Audio Processing, Vol.9, No. 8, pp.914-931, November 2001

[16] Moore, B.C.J, "An Introduction to the Psychology of Hearing," Academic Press, 1997

[17] Moelants, D., Rampazzo, C., "A Computer System for the Automatic Detection of Perceptual Onsets in a Musical Signal," In Camurri, Antonio (Ed.) " KANSEI, The Technology of Emotion," pp. 140-146, Genova, 1997

[18] David, H.A, "The Method of Paired Comparison," Oxford University Press, 1988

[19] Hynninen, J., Zacharov, N., "Guineapig – a generic subjective test system for multichannel audio," in Proc. of the AES 106th Int. Conv., Audio Eng. Soc., 1999

[20] Moore, B.C.J, Glasberg, B.R., Baer, T., "A Model for the Prediction of Thresholds, Loudness and Partial Loudess," Journal of the Audio Engineering Society, Vol. 45, No. 4, pp.224, 1997

[21] Kylliäinen, M., Helimäki, H., Zacharov, N., Cozens, J., "Compact High Performance Listening Spaces," Euronoise, Naples, 2003

[22] Thurstone, L.L., "A Law of Comparative Judgement," Psychological Review, No. 34, pp.273-386, 1927

[23] Nunnally, J.C., Bernstein, I.H., "Psychometric Theory," McGraw-Hill, 1994

[24] Johnston, J.D., "Transform Coding of Audio Signals Using Perceptual Noise Criteria," IEEE Journal on Selected Areas in Communications, Vol.6, No.2, pp.314-323, February 1988

[25] Vetro, A., Sun, H., Wang, Y., "Object-Based Transcoding for Adaptable Video Content Delivery," IEEE Transactions on Circuits and Systems for Video Technology, Vol.11, No. 3, pp.387-401, March 2001

[26] Tan, W., Zakhor, A., "Video Multicast Using Layered FEC and Scalable Compression," IEEE Transactions on Circuits and Systems for Video Technology, Vol.11, No. 3, pp.373-386, March 2001